LA-UR-83-3423

CONF-831254--2

LA-UR--83-3423

DE84 003820

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: DIGITAL FILTERING IN A DISEASE DETECTION SYSTEM

AUTHOR(S): R. R. Brown

SUBMITTED TO: American Society of Agricultural Engineers, Chicago, IL, December 12, 1983.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DIGITAL FILTERING IN A DISEASE DETECTION SYSTEM

by

R. R. Brown

ABSTRACT

A low-pass, nonrecursive digital filter has been designed to process data in an automated enzyme immunoassay system. The sof'ware implemented filter has been installed in the Intel 80/10 system controller. A low-speed sample rate of three samples per second allowed the filter algorithm to be programmed in the high-level FORTRAN language with a resultant execution speed of 0.6 seconds per data array.

Fourier techniques are used to derive a zero phase shift filter algorithm from a frequency domain prototype. The resulting alogrithm is modified by a Hamming window to reduce transients and Gibbs phenomenon oscillations. Observations on the effectiveness of the filter under full system operation indicate a 90% data recovery rate.

I. INTRODUCTION

Digital filtering is a powerful system component that can be used in a wide variety of information systems. In this case, digital filtering is applied to the recovery of noise-obscured data collected during the rapid serodiagnosis of infectious diseases in animals. The nonrecursive filter is implemented in software using standard commercially available products.

State-of-the-art software and hardware are being designed at Los Alamos to automate the enzyme immunoassay (EIA) used as a new method of detecting infectious diseases. Of primary concern to the funding agency, the United States Department of Agriculture, are production of a system that automates the diagnosis process and proof that the theory involved has practical applications. Collaboration with Technicon Corp. resulted in an EIA-modified system capable of analyzing 240 unique serum samples per hour.

In the final stage of processing, the Los Alamos-Technicon analyzer occasionally produces large excursions in the output data caused by unstable hydraulic conditions. The persistence of this problem resulted in the design of a nonrecursive digital filter using Fourier techniques. The low-speed, low-pass filter is implemented in software and run on the Intel 80/10 computer system that also controls the EIA hardware sequencing.

II. LOS ALAMOS-TECHNICON AUTOANALYZER II

At Los Alamos, we are studying the EIA for the diagnosis of infectious diseases in cattle.[1] Part of the research involves evaluation and enhancement of the Technicon automated EIA system. The system is a specialized analyzer used for rapid detection of infectious diseases, toxic agents, and low-molecular weight compounds.

The EIA system (Fig. 1) basically uses a primary sampler, an incubation sampler, a reagent pump, and a colorimeter to implement the EIA sequence. The system, designed to process 240 unique serum samples per hour, provides positive or negative test results for each sample and some statistical information. The fluids involved in the process are transported from one device to the next through flexible tubing.

A critical factor in system operation is the separation of samples in the tubing. Samples are separated by following each sample segment with a wash fluid segment continuously injected with air bubbles. The bubbles provide a scrubbing action, which cleans the tubing, thus providing an isolation medium from one sample to the next.

The EIA results in a resample fluid having an optical density proportional to the amount of infection present in the resample. At this point, the fluid is transported to the colorimeters, which produce an analogue voltage proportional to the optical density of the corresponding resample. A voltage-versus-time waveform generated at the colorimeter output (Fig. 2) is similar to a sinusoidal waveform with a peak occurring every 30 seconds.

A problem arises at the colorimeter because the resample must be completely devoid of bubbles. The colorimeter contains a debubbler to remove the bubbles before analysis. Occasionally, the debubbler fails to remove a bubble, which then obscures the data with a voltage spike (Fig. 3). The dominant Fourier series spectral components are significantly higher for the spike then for the resample response. Thus, data obscured by a bubble spike can be reclaimed if the waveform is passed through a low-pass filter.

The existing system data collection structure allowed easy incorporation of either an analogue filter or a software-implemented digital filter.

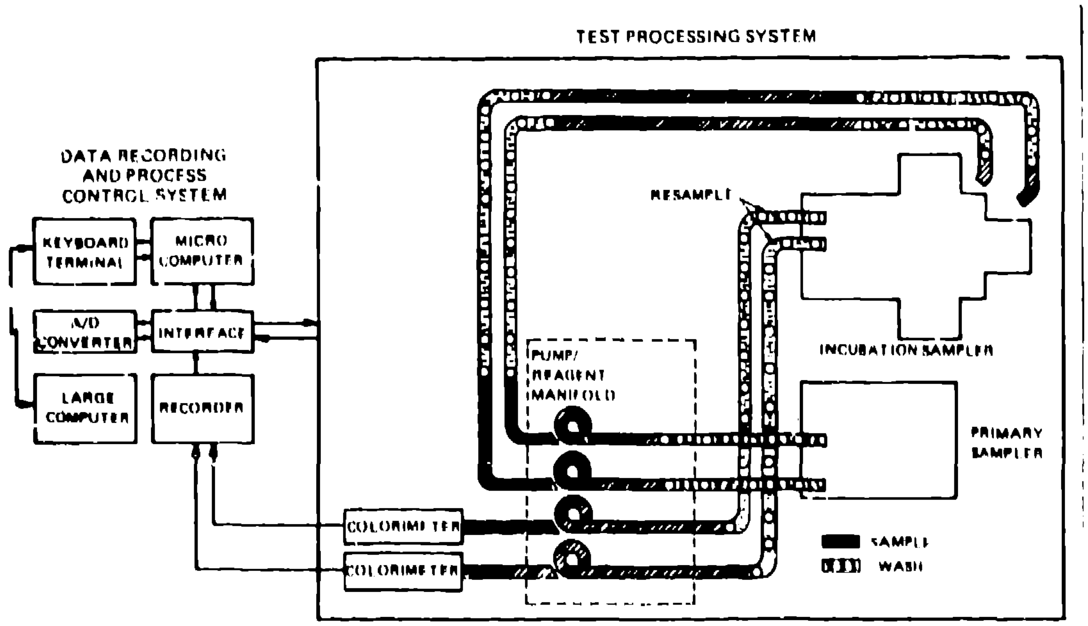


Fig. 1.

Diagram of the EIA system detailing the sample fluid isolation while in transport between modules.

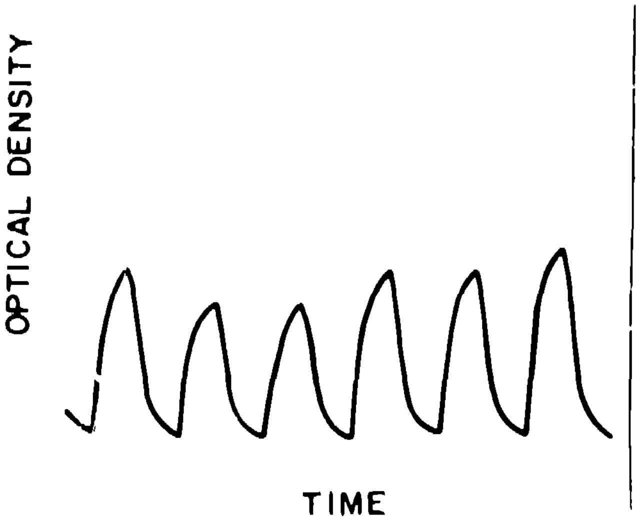


Fig. 2.
Actual waveforms produced as the resample segments pass through the colorimeter (six segments shown).

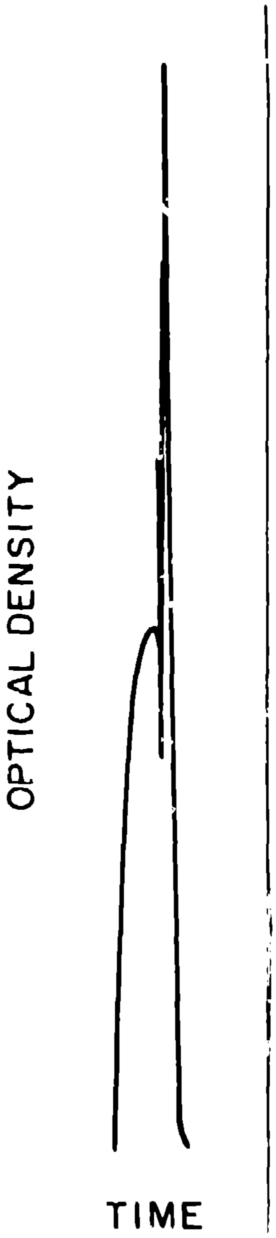


Fig. 3.
Spike response caused by the inclusion of a bubble in the resample segment in the postdebubbler section of the colorimeter.

The colorimeter data are collected both on a strip chart recorder and by the system computer. The computer digitizes the data at a rate of 3.08 samples per second. For each resample response, 40 data points are digitized, stored, and processed in real time. The processing algorithm determines the peak value of the resample response waveform, which is the desired end result of the EIA.

III. FILTER DESIGN

Nonrecursive filters$^{2-4}$ are unusual in that zero or constant phase shift can be achieved and the problem of stability doesn't arise. A nonrecursive filter is absolutely stable because there are no poles in the denominator.

The basic steps involved in implementing the digital filter are specification of filter frequency response and data, choice of filter methods and filter window (if any), and installation of a filter in the system.

A. Frequency Response

The ideal low-pass filter response (Fig. 4) absolutely excludes frequency components above the desired cutoff frequency ω_c while passing frequency components below ω_c without change. These characteristics are represented by the frequency domain transfer function

$$H(\omega) = \begin{cases} 1, \ \omega \leq \omega_c & \text{(passband),} \\ 0, \ \omega < \omega_c & \text{(stopband).} \end{cases} \qquad (1)$$

Although the ideal filter is used in generating the filter algorithm, the resultant digital filter only approximates the ideal response. In this case, the transition region between the passband and the stopband, the passband variations, and the stopband attenuation all must be specified as dictated by the application. Then iterative methods are used to generate the corresponding filter transfer function.

When choosing $_c$, the Nyquist frequency must be calculated by Eq.(2) to determine the upper limit on the frequencies that can be dependably filtered.

$$f_N = \frac{1}{2\,T}\ , \quad T = \text{sample rate.} \qquad (2)$$

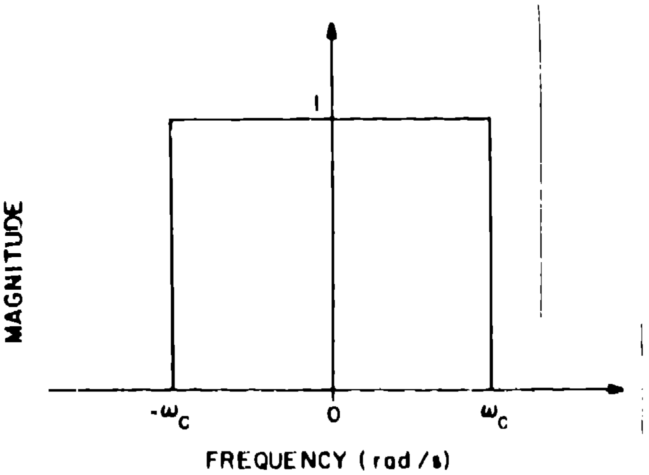


Fig. 4.
Ideal low-pass filter transfer function vs frequency plot with a cutoff frequency of ω_c.

If the data being filtered have a spectral content above the Nyquist limit, an analogue low-pass filter probably will be necessary to exclude frequencies above the Nyquist limit. Prefiltering is necessary because the filter transfer function repeats itself at intervals of twice the Nyquist limit.

B. Fourier Techniques

Fourier-series techniques[3-7] for nonrecursive filter design are useful in generating real-time filter algorithms from a known desired filter transfer function. The desired Z-domain transfer function is of the form

$$H(Z) = \left[\sum_{k=-J}^{k=J} \beta_k Z^k\right] Z^{-n} ,$$

where Z^{-n} is the time delay factor. When $Z = e^{j\ t}$ is substituted, the corresponding frequency response is

$$H(\omega) = \left[\sum_{k=-J}^{k=J} \beta_k e^{jk\omega T}\right] e^{-jn\omega T} = \sum_{k=-J}^{k=j} \beta_k e^{j(k-n)\omega T} . \qquad (3)$$

By expanding Eq.(3) using the Fourier-series approximation, the coefficients are determined for either even H( ) = H (-) or odd H ( ) = -H(-) desired transfer functions

$$a_k - \beta_k = \frac{2T}{\pi}\int_0^{\omega_c} H_D(\omega)\cos(k\omega T)\, d\omega \quad ; \quad \text{(even)} \qquad (4)$$

$$a_o = \beta_o = \frac{T}{\pi}\int_0^{\omega_c} H_D(\omega)\, d\omega \quad ; \quad \text{and} \qquad (5)$$

$$b_k = \beta_k = \frac{2T}{\pi}\int_0^{\omega_c} H_D(\omega)\sin(k\omega T)\, d\omega \quad , \quad \text{(odd)}$$

where $H_D($ $)$ is the desired transfer function and $_c$ is its upper cutoff frequency. The general equation for the desired transfer function is

$$H_D(Z) = a_o + \frac{1}{2}\left[\sum_{k=1}^{\infty}(a_k - b_k) Z^{-k} + \sum_{k=1}^{\infty}(a_k + b_k) Z^k\right] . \qquad (6)$$

In the case of the low-pass filter of Fig. 4, the transfer function is an even function, and thus Eq.(6) becomes

$$H_D(Z) = a_o + \frac{1}{2}\left[\sum_{k=1}^{\infty} a_k Z^{-k} + \sum_{k=1}^{\infty} a_k Z^k\right] . \qquad (7)$$

Note that the upper limit on the summation in Eq.(7) has a practical limit dictated by the hardware used to implement the filter and the time constraint placed on the hardware. Lowering the summation upper limit Eq.(8) makes the resulting filter algorithm smaller (with a faster hardware execution time) but also makes approximation of the desired filter frequency response cruder.

$$H_D(Z) \cong a_o + \frac{1}{2}\left[\sum_{k=1}^{J} a_k Z^{-k} + \sum_{k=1}^{J} a_k Z^k\right] \quad , J < \infty \quad . \qquad (8)$$

If the allowable transition region is known from the passband to the stopband of the frequency domain filter transfer function, then J can be calculated from J ≈ 2 /(T), where =($_2$ - $_1$)/2. (See Fig. 5.)

C. Time Domain Mapping

The inverse Z transformation of Eq.(8) is the time domain representation of the desired digital filter. Equation (9) is the inverse Z transformation of the transfer function of Eq.(8) where Y is the output and X is the input of the digital filter. (Note that it is implicit that the discrete-time variables are multiplied by the sampling rate constant T.)

$$Y(n) = a_0 + \frac{1}{2}\left[\sum_{k=1}^{J} a_k X(n-k) + \sum_{k=1}^{J} a_k X(n+k)\right] . \qquad (9)$$

Equation (9) is a discrete time algorithm in a form suitable for installing into the target system. In this equation, the filtering is done by the weighted summation of past, present, and future sampled data points.

The target system computation time per data point can be calculated by

$$t = 2J\,t_m + (2J+1)\,t_A + (4J+1)\,t_{f-s} \ , \qquad (10)$$

where t_m = multiplication execution time, t_A = addition execution time, t_{f-s} = memory fetch-store pair time, and J = summation limit on Eq.(9).

D. Filter Window

Windowing is a process of selective attenuation used in digital filtering to reduce transients in the time domain filtered data. Start-up and ending transients (or leakage) are due primarily to an aperiodic filter input; they can be greatly reduced by windowing either the filter coefficients or the data. Although a window improves a filter's transient response, it also degrades some desirable filter characteristics. For example, windowing lessens the slope of the frequency domain transfer functions transition region.

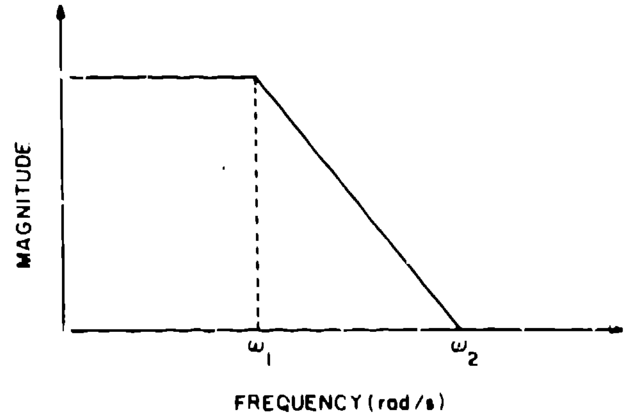


Fig. 5.
The low-pass filter transfer function plot shows the transition region between the passband and stopband as being $\omega_2 - \omega_1$.

The application dictates the type of window used. The type often depends on how long the filter algorithm or data array is and what sort of analysis is to be done on the data. If the data are to be analyzed for spectral content, often windowing is done on the data. Because the window is basically a position-selective attenuator, the data spectral content may not change (depending on the length and range of the data array) except that the leakage problem is reduced. If data are to be analyzed for time domain amplitude, windowing the data will most likely be prohibitive because of the window's attenuation characteristics. In this situation, windowing can be done on ~~the filter algorithm. When Fourier techniques are used for~~ filter design, an added advantage to windowing the filter algorithm is the reduction of the Gibbs phenomenon oscillation in the filter frequency response.

IV. LOW-PASS FILTER APPLICATION

The digital filter resides in the Los Alamos-Technicon EIA analyzer's controller. The controller first digitizes the colorimeter output and stores the results in an array. The array then is processed by the software-implemented digital filter algorithm and finally is analyzed for the peak value. The peak value is corrected to an optical density, corresponding to the particular serum sample.

Forty data points are digitized per serum sample at a rate of 0.325 seconds per sample (= T). The Nyquist frequency is then

$$f_N = \frac{1}{2\,T} \cong 1.5 \text{ Hz.}$$

The roughly sinusoidal shape of the colorimeter data (Fig. 6) yields an approximate calculation of the filter passband frequencies,

$$f_{data} \cong \frac{1 \text{ cycle}}{30 \text{ seconds}} = 0.0333 \text{ Hz.}$$

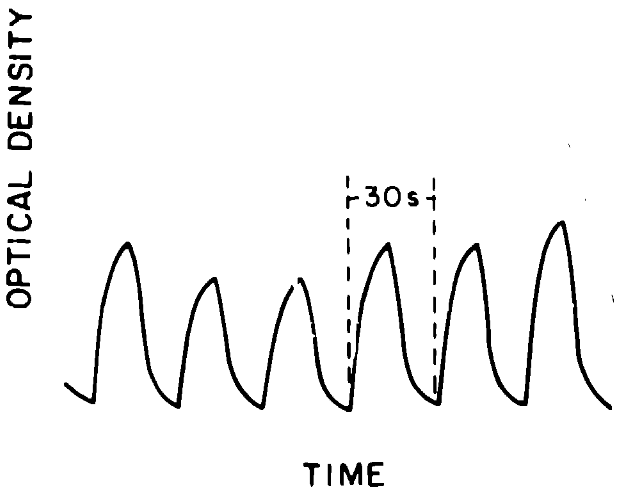


Fig. 6.
A resample segment produces a colorimeter response that is approximately sinusoidal with a period of 30 seconds.

By arbitrarily choosing the cutoff frequency as five times the data frequency, the cutoff frequency is

$$f_c \simeq 5\, f_{data} \simeq 0.2 \text{ Hz.}$$

Figure 7 is the desired frequency domain transfer function of the low-pass filter. Because the low-pass filter repeats itself at intervals of twice the Nyquist limit ($2\ f_N$=3.0 Hz), the equivalent of a bandpass filter will be from 2.8 to 3.2 Hz (and again from 5.8 to 6.2 Hz, etc.). Thus, the colorimeter output must be analogue prefiltered if the colorimeter output contains a spectral content above 2.8 Hz. In this case, an analogue low-pass filter with a cutoff frequency of 2.8 Hz could be placed at the colorimeter output. (A simple one- or two-pole filter will do.)

Generating the digital filter algorithm is a matter of releting the desired frequency response Eq.(1) to the Fourier technique equations Eqs.(4) and (5).

$$a_k = \frac{2T}{\pi}\int_0^{\omega_c} H_D(\omega)\cos(k\omega T)\,d\omega = \frac{2T}{\pi}\int_0^{2\pi f_c}(1)\cos(k\omega T)\,d\omega = \frac{2}{k\pi}\sin(2\pi k f_c T)\quad k = 1, 2, 3, \ldots J,$$

and

$$a_0 = \frac{T}{\pi}\int_0^{\omega_c} H_D(\omega)\,d\omega = \frac{T}{\pi}\int_0^{\omega_c}(1)\,d\omega = 2 f_c T$$

where the number of filter algorithm taps is J + 1 for a casual algorithm. For the noncasual symmetrical algorithm,

$$H(Z) = a_0 Z^J \ldots a_{J-1} Z + a_J + a_{J+1} Z \ldots a_{2J} Z^{-J}, \tag{11}$$

the number of filter taps = 2 J + 1. Equation 11 is a nonrecursive digital filter with zero phase shift and a time domain form of $Y(n) = a_0X(n + J) \ldots a_{J-1}X(n) + a_J + a_{J+1}X(n - 1) \ldots a_{2J}X(n - J)$, where $X(n + J)$ = input at time $(n + J)T$, and $Y(n)$ = output at time nT. The tap length of the filter is proportional to the accuracy by which the desired transfer function (Fig. 7) is reproduced. Of primary concern is the frequency domain transfer function passband-to-stopband transition region. The effect of the tap length can be seen in Fig. 8 and Fig. 9. Figure 8 is an 11-tap filter with a transition region of 140 dB/DECADE; Fig. 9 is a 41-tap filter with a transition region of nearly 170 dB/DECADE. Both filters were generated from the same set of parameters.

The frequency response of a filter derived from Fourier techniques always exhibits Gibbs' phenomenon oscillation (Fig. 10) at any discontinuity. Modifying the filter transfer function can greatly reduce Gibbs' phenomenon and the time domain start-up and ending transients. The modification is a window function

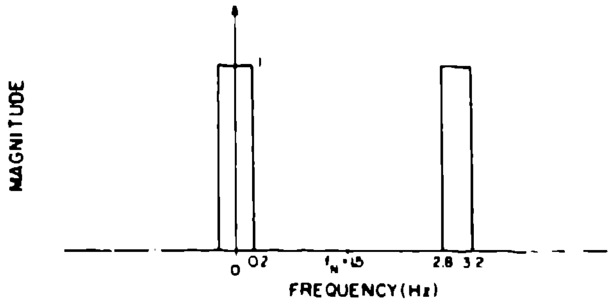


Fig. 7.
A plot of low-pass filter transfer function showing frequency folding effects.

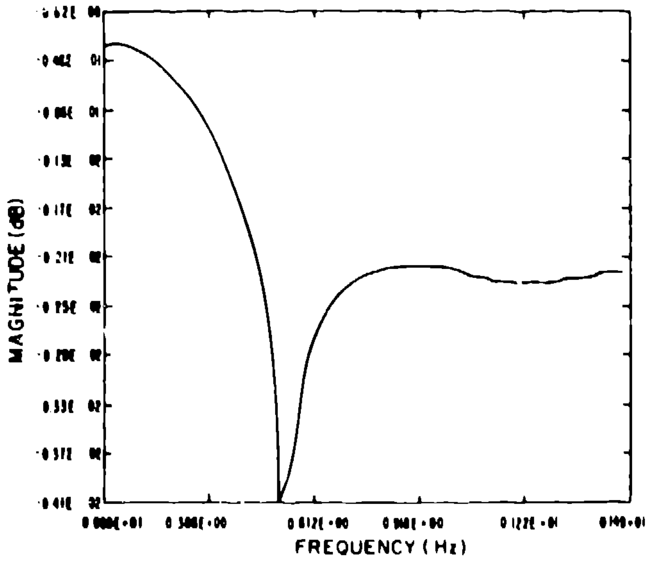


Fig. 8.
A plot of an 11-tap filter.

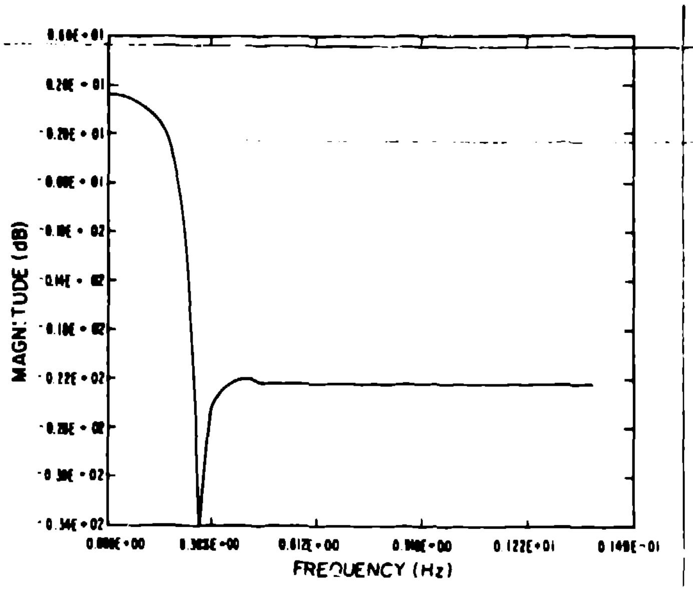


Fig. 9.
A plot of 41-tap filter.

in which each filter coefficient Eq. (11) is multiplied by a corresponding window coefficient. A good window for a relatively small number of terms is the Hamming window,[8]

$$W(k) = 0.54 - 0.46 \cos\left(\frac{\pi k}{J}\right) \quad , -J \le k \le J$$

Figure 11 shows the windowed version of the filter in which the Gibbs' phenomenon oscillation is essentially eliminated. Figure 12a and b show simulation of the time domain sampled data before and after filtering. In Fig. 12a, the triangle curve is the colorimeter output sampled data, while the dot curve is the data after filtering without a window. In Fig. 12b, the corresponding windowed filtering exhibits a transient behaviour lower than that in Fig. 12a; it is due solely to the modification of the filter transfer function by the Hamming window.

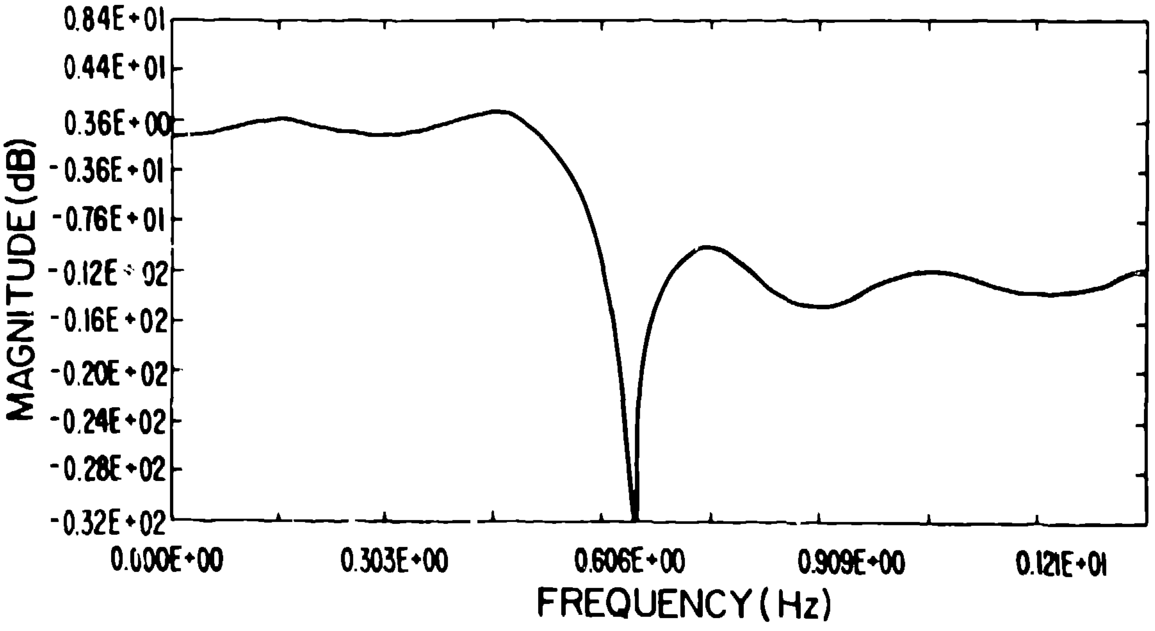


Fig. 10.
A plot of a filter transfer function showing Gibbs' phenomenon oscillations.

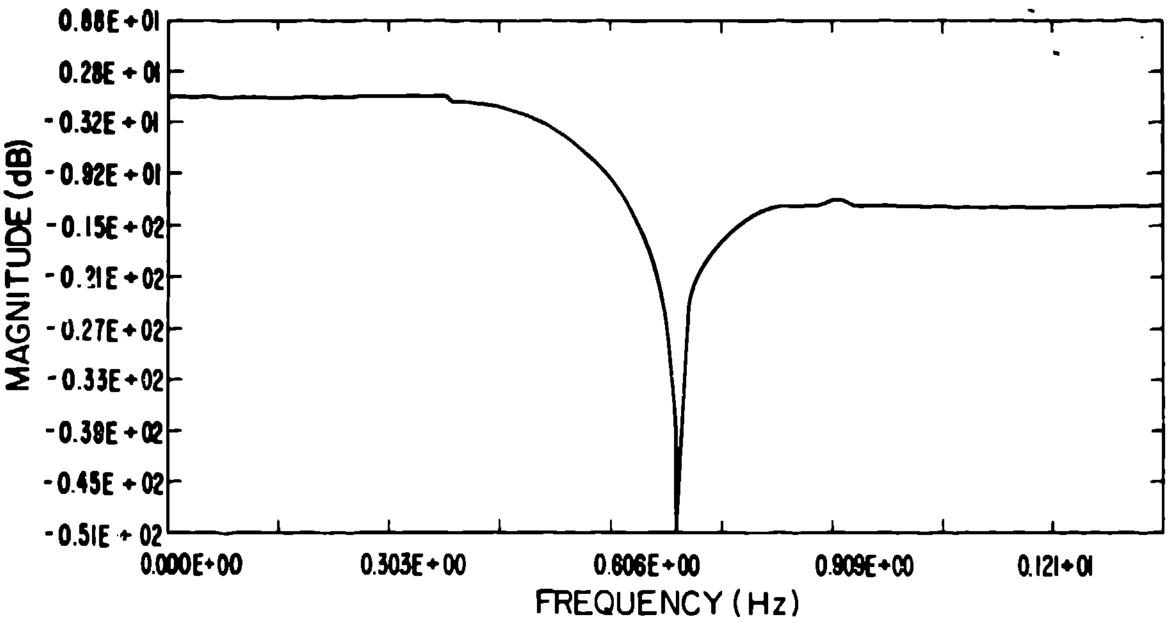


Fig. 11.
A display of the transfer function of Fig. 10 plotted after windowing.

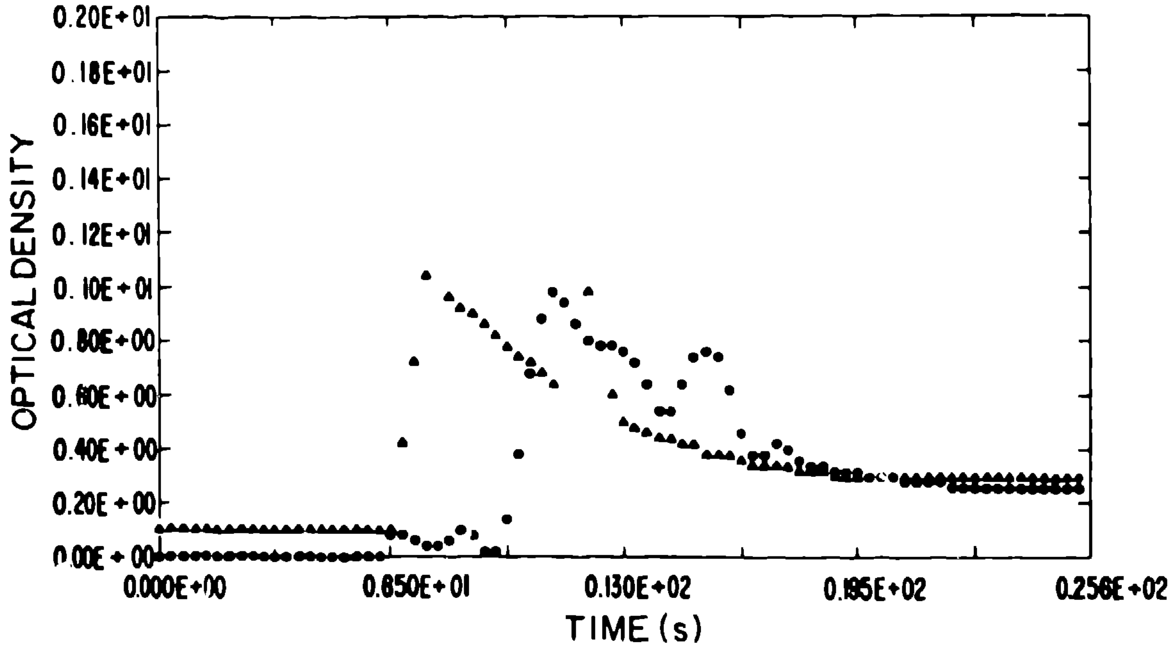


Fig. 12a.
The time domain plot displays the difference between the sampled data with (dot curve) and without (triangle curve) filtering.

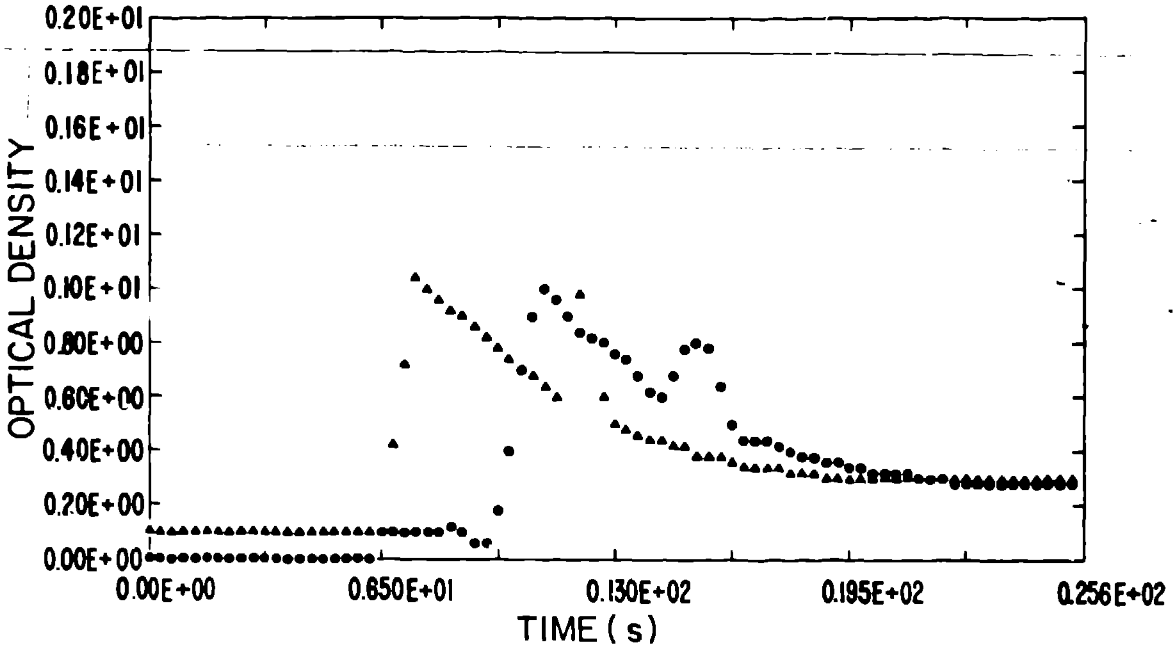


Fig. 12b.
Plot of the same data and filter as in Fig. 12a, except that the dot curve is the result of a windowed filter.

When the filter algorithm is complete, software is written to implement the filter in the Technicon controller. An Intel 80/10 computer with the SBC 310 high-speed mathematics unit controls the system. The controller is programmed with approximately 8000 lines of PLM source code and a small amount of FORTRAN 80 code. Because of the precision involved in the filter processing and the ease of interfacing to the SBC 310, the FORTRAN 80 language is advantageous in implementing the digital filter. The Appendix lists the 30 lines of FORTRAN required to implement the filter.

The filter tap length is dictated by the time allowable for completing the filter subroutine. In this case, a 21-tap filter requires 0.6 seconds of computer time to execute the routine. According to Eq.(10), the computer must execute 1200 floating-point multiplications, 1260 floating-point additions, and 2460 memory fetch-store pairs to process the 60 data points through the filter algorithm.

The 21-tap digital filter chosen for use in the Technicon system has a stopband of 22 dB and a transition region of about 160 dB/DECADE, as shown in Fig. 13. The filter is roughly equivalent to an eighth-order analogue filter.

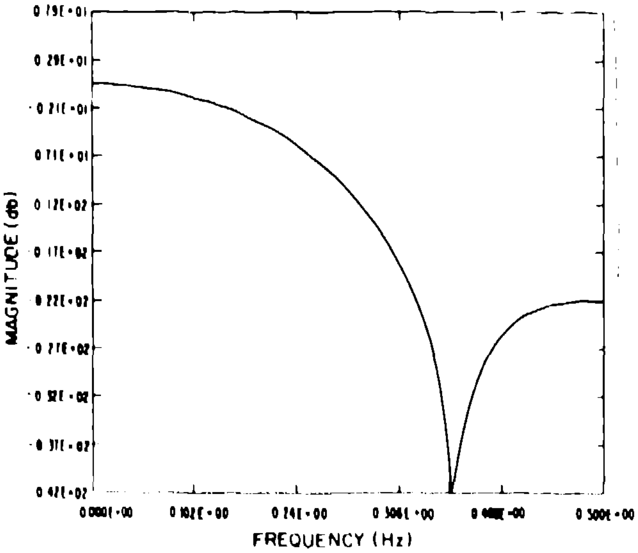
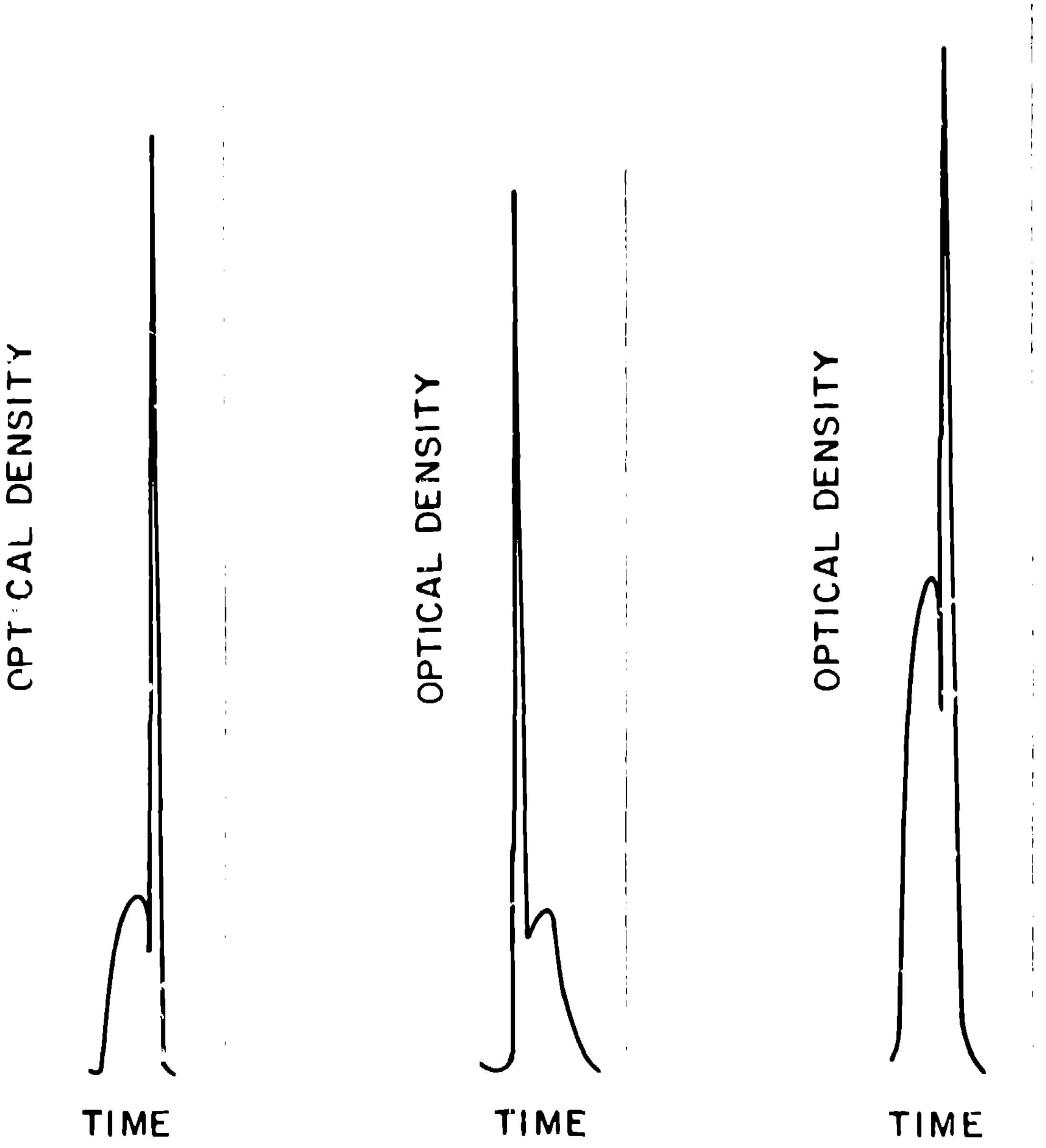


Fig. 13.
Transfer function of the filter algorithm used in the Technicon system.

To evaluate the filter's effectiveness, several EIA runs were made in which bubble-induced noise spikes were observed and the corresponding computed peak values noted. During a run, several quality control assays are included in which each computed peak value should be approximately equivalent. Figure 14 presents three typical sample responses with bubble-induced spikes that occurred during the quality control assays.

Included below are the following data values associated with Fig. 14; the overall assay mean value, the standard deviation, and the computer-calculated peak value of the sample response. An effective filter will process the data so that the presence of a spike will not affect the computer-calculated peak value; the peak value should be approximately the same as the assay mean value (within the given standard deviation). The data below verifies the filter's effective recovery of sample data obscured by bubble-induced noise spikes.

Fig. 14	Mean Value	Standard Deviation	Peak Value
a.	0.0062	0.0010	0.0066
b.	0.0062	0.0010	0.00:8
c.	0.0214	0.0020	0.0214



Figs. 14a, b. & c.
Colorimeterer Output with bubble-induced spikes for three resample segments.

The process of selecting the appropriate digital filter parameters for a given application is iterative. The tap length, the inclusion of windowing, the cutoff frequency, and the overall adjustment of the filter are all difficult to calculate effectively without a certain amount of iteration. Therefore, programming the filter design process by Fourier techniques has advantages. The user specifies the filter parameters then the computer calculates the filter coefficients. In addition, plots of transfer function magnitude in decibels versus frequency and a real-time simulation are generated. The simulation function accepts a user-defined data array then filters the data so the user can immediately see the effectiveness of the filter just created.

V. CONCLUSION

The use of digital filtering to recover noise-obscured data has many possible applications in information systems. The EIA system provides an ideal application for digital filtering because of sufficient separation of noise and data spectral composition. In addition, the slow data collection rate associated with the EIA allows a quick and flexible filter implementation in software.

The low-pass, nonrecursive filter design is a step-by-step process involving a choice of filter algorithm generating and windowing methods. Fourier techniques for algorithm generation provide the advantage of zero phase shifts, but also the disadvantage of Gibbs' phenomenon oscillations. The oscillations can be essentially eliminated by windowing the filter coefficients.

There are two primary reasons for using a nonrecursive filter in this application. A nonrecursive filter is simpler to design because the question of stability does not have to be addressed. The penalty paid is the longer execution time of the filter algorithm, which is affordable in this application. The transient response problem is less in nonrecursive filters than in recursive filters; small sampled data arrays benefit from a decreased transient response.

The digital low-pass filter successfully separates colorimeter serum data from bubble-induced spikes using a 21-tap algorithm. The filter algorithm and its associated array manipulations require 0.6 seconds of Intel 80/10 computing time (for 60 data points); the filter is approximately equivalent to an eight-pole analogue filter with a 22-dB stopband.

REFERENCES

1. Gonzalez, R. C. and Wintz, P. 1977. Digital Image Processing. Addison-Wesley, Reading, Massachusetts.

2. Hamming, R. W. 1977. Digital Filters. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. Sec. 13.3.

3. Oppenheim, A. V. 1978. Applications of Digital Signal Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

4. Oppenheim, A. V. and Schafer, R. W. 1975. Digital Signal Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

5. Papoulis, A. 1977. Signal Analysis. McGraw-Hill Book Co., New York.

6. Rabiner, L. R. and Gold, B. 1975. Theory and Application of Digital Signal Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

7. Seawright, G. L., Sanders, W. M. and Bryson, M. 1980. Automation of the Enzyme Immunoassay for the Serodiagnosis of Infectious Diseases in Livestock. The Ruminant Immune System. Plenum Press, New York and London: Vol. 137.

8. Sterns, S. D. 1975. Digital Signal Analysis. Hayden Book Company, Inc., Rochelle Park, New Jersey: pp. 104-109, 115-118.

9. Wait, J. V. and Korn, G. A. 1978. Digital Continuous - Systems Simulation. Prentice-Hall, Inc., Englewood Cliffs, New Jersey: pp. 239-241.